

# 计数dp

2019年3月18日 19:44

## 1. 将n个相同的球放进k个相同的盒，不能有空

- i. 引：不能有空=至少一盒只有1个球+每盒都超过1个球
- ii. 引：至少一盒只有1个球= $n-1$ 球放入 $k-1$ 盒
- iii. 引：每盒都超过1个球= $n-k$ 球放入 $k$ 盒
- iv. 初值：只要 $\text{ans}[0][0]=1$ 就能转移出所有把 $k$ 球放入 $k$ 盒的方法数=1了
- v. 

```
for__(i,1,n)
    for__(j,1,min(i,k))
        ans[i][j]= ans[i-1][j-1]+ ans[i-j][j];
```
- vi. 应用：整数 $n$ 拆成 $k$ 个数，115, 151, 511视作同一种拆法
- vii. 对应[第一行](#)的模型

## 2. 将n个相同的球放进k个相同的盒，能有空

- i. 引：能有空=至少一盒空+每盒都不空
  - ii. 引：至少一盒空= $n$ 球放入 $k-1$ 盒
  - iii. 引：每盒都不空= $n-k$ 球放入 $k$ 盒
  - iv. 初值： $\text{ans}[0][0]=1$ ，注意这里要把第0行都赋为1
  - v. 

```
for__(i,0,n)
    for__(j,1,k)
        if(i>=j)
            ans[i][j]= ans[i][j-1]+ ans[i-j][j];
        else
            ans[i][j]= ans[i][j-1];
```
  - vi. 应用：此时的 $\text{ans}[n-m][m]$ 即为前一题的 $\text{ans}[n][m]$
  - vii. 对应[第二行](#)的模型
  - viii. 经手动test2，确认上一题答案的前 $m$ 项和即为本题的答案
- ```
const int MN = 1e3 + 5;
const int P = 1e3;
bool show = 0;
int n,k;
ll ans1[MN][MN],ans2[MN][MN];
ll sum1[MN][MN];

void init(){
    ans1[0][0]= ans2[0][0]= 1;
    for__(i,0,n) for__(j,1,i)
        ans1[i][j]= ans1[i-1][j-1]+ ans1[i-j][j], ans1[i][j] %= P;
    for__(i,0,n) for__(j,1,k)
        if(i>=j) ans2[i][j]= ans2[i][j-1]+ ans2[i-j][j], ans2[i][j]%= P;
        else ans2[i][j]= ans2[i][j-1], ans2[i][j]%= P;
    for__(i,1,n) for__(j,1,i)
        sum1[i][j]= sum1[i][j-1] + ans1[i][j], sum1[i][j]%= P;}

void test1(){
    puts("testing !!");
    if(show){
        for__(i,1,n) for__(j,1,i)
            printf("%4lld%c",ans1[i][j],"\n"[j==i]);
        for__(i,1,n) for__(j,1,i)
```

```

        printf("%4lld%c",ans2[i-j][j]," \n"[j==i]);
    for__(i,1,n) for__(j,1,i)
        if(ans1[i][j] != ans2[i-j][j]) printf("%d %d ",i,j), puts("!!!!!!");}
void test2(){
    puts("testing ??");
    if(show){
        for__(i,1,n) for__(j,1,i)
            printf("%4lld%c",sum1[i][j]," \n"[j==i]);
        for__(i,1,n) for__(j,1,i)
            printf("%4lld%c",ans2[i][j]," \n"[j==i]);}
    for__(i,1,n) for__(j,1,i)
        if(sum1[i][j] != ans2[i][j]) printf("%d %d ",i,j), puts("??????");}

int main(int argc, char** argv) {
    k = n = 1000;
    init();
    test1();
    test2();
    return 0;}

```